- 1 -

DESCRIPTION

PSEUDO-OVERWRITING DATA ON WRITE-ONCE DISCS

TECHNICAL FIELD

The present invention relates to a recording method for a write-once disc using a logical overwritable mechanism and a semiconductor integrated circuit for use in a recording apparatus and/or a reproduction apparatus.

BACKGROUD ART

File systems for optical discs have made advances through various activities to develop UDF (Universal Disk Format®) specifications published from OSTA (Optical Storage Technology Association).

For write-once discs, the recording method has improved from multi-session recording to file-by-file recording using VAT (Virtual Allocation Table).

On the other hand, for rewritable discs, a volume and file structure has improved from the structure using non-sequential recording defined in ECMA 167, which is the international standard, to the structure using Metadata Partition specified in UDF Revision 2.5 (hereinafter UDF 2.5). The merits to use Metadata Partition are the improvement in the performance to retrieve metadata, such as file entries/directories, and to increase the robustness from media damage.

However, Metadata Partition cannot be used for data appending usage on a write-once disc. This is because it is not allowed in UDF 2.5 to use Metadata Partition with VAT, due to the difficulty to implement this

- 2 -

combination.

Typically, it is also difficult to develop a new recording method for a write-once disc. This comes from the physical characteristics such as that the data written at once can not be overwritten, hence it would be required to study from the several aspects, to be consistent with computer architecture, the possibility of implementation for drive apparatus, the restrictions due to the dedicated resource of consumer appliances, etc.

The present invention has been made in view of the above subjects and includes an objective of providing the merits of Metadata Partition to the data recording usage on a write-once disc.

DISCLOSURE OF THE INVENTION

A recording method for instructing a drive apparatus having a pseudo-overwrite function to write data on a write-once disc according to the present invention includes: (a) receiving a write request which specifies at least data for a file to be written; (b) instructing the drive apparatus to read metadata for managing the file from a location in the write-once disc, so as to obtain the metadata; (c) querying a next writable address indicating a location at which data is to be written next to the drive apparatus, so as to obtain the next writable address; (d) updating the metadata to reflect the writing of the data specified by the write request; (e) instructing the drive apparatus to write the data specified by the write request to a location indicated by the next writable address in the write-once disc; and (f) instructing the drive apparatus to write at least a part of the updated metadata to the location from which the metadata is read in the step (b) in the write-once disc.

In one embodiment of the present invention, the steps (e) and (f) are performed using the same write instruction.

In one embodiment of the present invention, the step (f) is performed after the step (e) is performed.

In one embodiment of the present invention, the updated metadata includes a file entry of a directory under which the file is recorded.

In one embodiment of the present invention, the updated metadata includes a file entry of the file.

According to another aspect of the present invention, a system controller is provided for instructing a drive apparatus having a pseudo-overwrite function to write data on a write-once disc, the system controller including a controller for controlling the drive apparatus, wherein the controller is configured to perform a process including the steps of:(a) receiving a write request which specifies at least data for a file to be written; (b) instructing the drive apparatus to read metadata for managing the file from a location in the write-once disc, so as to obtain the metadata; (c) querying a next writable address indicating a location at which data is to be written next to the drive apparatus, so as to obtain the next writable address; (d) updating the metadata to reflect the writing of the data specified by the write request; (e) instructing the drive apparatus to write the data specified by the write request to a location indicated by the next writable address in the write-once disc; and (f) instructing the drive apparatus to write at least a part of the updated metadata to the location from which the metadata is read in the step (b) in the write-once disc.

In one embodiment of the present invention, the controller includes a semiconductor integrated circuit.

According to another aspect of the present invention a program is provided for use in a system controller for instructing a drive apparatus having a pseudo-overwrite function to write data on a write-once disc, wherein the program is configured to perform a process including the steps

of: (a) receiving a write request which specifies at least data for a file to be
written; (b) instructing the drive apparatus to read metadata for managing
the file from a location in the write-once disc, so as to obtain the metadata;
(c) querying a next writable address indicating a location at which data is to
5   be written next to the drive apparatus, so as to obtain the next writable
address; (d) updating the metadata to reflect the writing of the data
specified by the write request; (e) instructing the drive apparatus to write
the data specified by the write request to a location indicated by the next
writable address in the write-once disc; and (f) instructing the drive
10   apparatus to write at least a part of the updated metadata to the location
from which the metadata is read in the step (b) in the write-once disc.

These and other advantages of the present invention will become
apparent to those skilled in the art upon reading and understanding the
15   following detailed description with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

20   Figure 1 is a diagram illustrating a configuration of areas when a
file is recorded.

Figure 2 is a diagram illustrating a configuration of areas when a
Data-A file is recorded in a root directory of a disc having the state as
25   shown Figure 1.

Figure 3 is a diagram illustrating a configuration of areas when a
Data-A file is recorded in a root directory on a disc which having a larger
spare area.

30
Figure 4 is a flowchart illustrating a procedure for recording a file.

Figure 5 is a diagram illustrating an optical disc information

recording/reproduction system.

Figure 6 is a diagram illustrating transfer of commands between a drive apparatus and a system controller.

Figures 7A-7D each comprise a diagram illustrating the blocks in a user data area, when the data is remapped for a rewritable disc case and write-once disc case.

Figures 8A and 8B each comprise a diagram illustrating the blocks in a user data area, when the data is written to NWA.

Figure 9 is a diagram illustrating the data structure of the remapping table.

Figure 10 is a diagram illustrating a configuration of areas to explain a pseudo-overwrite method.

Figures 11A and 11B each comprise a flowchart illustrating a procedure to write the data by an optical disc information recording/reproduction system.

Figure 12 is a diagram illustrating an optical disc information recording/reproduction system which is a part of consumer video recorder or consumer video player.

Figure 13 is a flowchart illustrating a procedure to write the data on a write-once optical disc by the recording/reproduction system explained in Figure 12.

Figure 14 is a flowchart illustrating a procedure to read the data from a write-once optical disc by the recording/reproduction system explained in Figures 5 and 12.

- 6 -

Figure **15A-15D** each comprise a diagram illustrating the blocks in a user data area, when the data is written using a pseudo-overwrite method.

5

Figures **16A-16C** each comprise an example of track layouts after logical format, and after some files are recorded.

10   BEST MODE FOR CARRYING OUT THE INVENTION

An overwritable function for a write-once disc performed by drive apparatus has been studied.   However, it was difficult to put it in practice, because the drive apparatus cannot know how much data will be

15   overwritten and where the data will be overwritten.

For example, the data to be overwritten is stored in the other location as a write-once disc, and the information to specify the original location and the replacement location have to be handled in the drive

20   apparatus.   When the amount of overwritten data increases, it takes a longer time to search where it is replaced.   Hence, such a drive apparatus could not read/write with enough performance due to its small resources (e.g. CPU speed and memory).

25   It is believed the new file system should distinguish the data to be overwritten and the data to be newly written, and the new file system can match with the drive apparatus which has an overwritable function.   Then, the possibility to apply Metadata Partition for data appending usage on a write-once disc, without using VAT is found.

30

The strategic importance of this idea includes; by having the device handle the overwriting of existing blocks, the file system does not need to implement the logic.   This reduces the complexity of the file

system driver.

In the following embodiments, the investigations based on this idea are shown in detail.

(Embodiment 1)

The amount of the overwritten data can be reduced by optimizing the procedure in the file system.　In this embodiment, the basic read/write operation is explained in accordance with the new recording method for the drive apparatus with the overwritable function of a write-once disc.

Hereinafter, embodiments of the present invention will be described with reference to the drawings.

Figures 1 and 2 are diagrams showing a configuration of areas. Figure 4 is a flowchart showing a procedure for recording a file.　Figure 1 shows the state where a file has been recorded after a logical format operation.　Figure 2 shows the state where a Data-A file has been recorded in a root directory on a disc having the state of Figure 1.

Firstly, Figure 1 will be described.

The data area includes the areas such as a Lead-in area, a volume space and a Lead-out area, (wherein the Lead-in area and the Lead-out area are managed by a physical layer).　The physical sector is an addressable unit in the data area and the physical sector number is assigned in ascending sequence to each physical sector.　The volume space consists of logical sectors and a logical sector number is assigned in ascending sequence to each logical sector.　Each logical sector corresponds to the physical sector uniquely in advance.　For example, the logical sector with a logical sector number 0 corresponds to the physical sector with physical sector number 10000 and the start address of the

- 8 -

volume space is stored in the Lead-in area.

The defect Management Area (DMA) is an area, in which information indicating the correspondence between the address of a block to be replaced and the address of a replaced block in a replacement operation, is recorded as a defect list.

Temporary DMA (TDMA) is an area, in which a temporary defect list is recorded in an incremental write operation. When a disc is finalized to prohibit incremental write operations, a temporary defect list is registered as a defect list in DMA. DMA is provided in two portions of a disc, i.e., at an inner portion and at an outer portion. Thus, a defect list is recorded in different areas twice. In DMA, disc information, such as track information, positional information of a spare area, and the like, are recorded.

The spare area is a replacement area, in which data is recorded by a replacement operation, which is equivalent to a linear replacement method. The spare area is assigned outside a volume space which is handled by a file system. In this example, data has been recorded in a portion of the spare area. Addresses in the spare area are inherently specified by using physical addresses. To simplify the explanation, relative addresses in the spare area are indicated by SA's (Spare area Address). Sectors on and after SA #m are in the unrecorded state.

In the present invention, the linear replacement algorithm which is generally used for defect management is applied to overwriting performed by the drive apparatus.

The volume space consists of three tracks. A track is an area in which the data is recorded sequentially from the beginning of the track on a write-once disc. The end of recorded area in a track is managed by a drive apparatus. Track Status (Close and Open) indicates a status of a track. "Close" indicates that all sectors in a track have been used for data

recording.   "Open" indicates that there is a sector(s), which has not been used for data recording.   In other words, data can be incrementally written into an open track.

5   (Volume Structure)

           The volume and file structure complies with UDF 2.5.   A volume structure, which is located at an area having a smaller logical sector number, includes Anchor Volume Descriptor Pointer, Volume Recognition

10   Sequence, Volume Descriptor Sequence, and Logical Volume Integrity Sequence.   A volume structure, which is located at an area having a larger logical sector number, includes Anchor Volume Descriptor Pointer and Volume Descriptor Sequence.   The Logical Volume Integrity Sequence, in which a Logical Volume Integrity Descriptor is recorded, is a part of a

15   volume structure.   However, for convenience of explanation in this example, the Logical Volume Integrity Descriptor is explicitly described under the volume structure.   Since the volume structure has been previously recorded, Tracks #1 and #3 are in the close state.   The area described as Track #2 is assigned as a partition specified by UDF.   The

20   Metadata file is also called a Metadata partition.   To distinguish it from the Metadata partition, Track #2 area is called a physical partition.   In the Metadata file, an unused area is recorded in advance. Track #2 is an area for recording file data.   Therefore, an area following the recorded area is in the unrecorded state.

25
           (File Structure)

           Metadata Bitmap FE (Metadata Bitmap file File Entry) is a file entry for organizing the areas allocated for a Metadata Bitmap.   Metadata

30   Bitmap is a bitmap for specifying available sectors which are ready for use in a Metadata file.   Not only unrecorded areas, but also an area which becomes an unused area by deleting a file entry or a directory, are registered in the bitmap as available areas.   Metadata file FE (Metadata

file File Entry) is a file entry for organizing the areas allocated for a
Metadata file.   In a Metadata file, file entries and directories are recorded.
In UDF, a File Set Descriptor is also recorded, which is not shown in the
figure.

5

Root directory FE (root directory file entry) is a file entry for
organizing the areas allocated for the root directory.   A root directory FE is
recorded in MA #i.   A root directory is recorded in MA #i+1.   Though not
shown, the root directory FE and the root directory are actually stored

10   physically in sectors in the spare area.   Information indicating which sector
in the spare area replaces a root directory FE and a root directory is
recorded in TDMA.   MA (Metadata file Address) indicates a relative
address within a Metadata file.   Since a file(s) have been previously
recorded, areas on and after MA #k are available.

15

(File recording procedure)

An exemplary procedure for recording a Data-A file onto the write-
once disc of Figure 1 will be described with reference to Figures 2 and 4.

20

In step S101, the Metadata Bitmap is read into a memory and is
updated in the memory to obtain a recording area in the Metadata file.

In step S102, a directory, under which a file is to be registered, is

25   read into the memory, and is updated in the memory.   In this example, the
root directory is read out, and the Data-A file is registered.

In step S103, the file entry of the directory is read into the memory,
and information (e.g., size and update time, etc.) of the directory is updated.

30

In step S104, the Data-A file data is recorded from the beginning
of the unrecorded area in Track #2.

- 11 -

In step **S105**, in order to register the positional information of the recorded data, the file entry of the file is generated in the memory.

In step **S106**, the data updated or generated in the memory is
5   recorded.   In the example of Figure 2, a drive apparatus is instructed to record the Metadata Bitmap file in the same place.   Since the specified area is an already-recorded area, the drive apparatus records data at SA #m, which is the beginning of the unrecorded area in the spare area.   It is instructed that the root directory is recorded at MA #k.   Therefore, the
10   root directory recorded at MA #i+1 becomes invalid, and the sector at MA #i+1 becomes an available sector in a logical space.   Even if it is instructed that data is recorded into an available area in the Metadata file, the data cannot be recorded into the area, since the all area in the Metadata file is already recorded in advance.   Therefore, the data of the
15   root directory is recorded into SA #m+1 in the spare area by a replacement operation.

The replacement operation is a pseudo-overwrite operation of the present invention.   As used herein, the term "pseudo-overwrite operation"
20   refers to a logical overwrite operation, in which the mechanism of a replacement operation is used to write the data into the unrecorded area in response to an instruction to write the data into an already recorded area.

It is instructed that the file entry of the root directory is written into
25   MA #i.   In this case, MA #i is an already-recorded area.   Therefore, the data is stored into SA #m+2 in the spare area by a pseudo-overwrite operation.   It is instructed that the file entry of the Data-A file is written into MA #k+1.   The data is stored into SA #m+3 by a pseudo-overwrite operation.
30

In step **S107**, it is instructed that the Logical Volume Integrity Descriptor is updated to indicate the integrity state of the file structure. The data is stored into SA #m+4 by a pseudo-overwrite operation.

As described in step **S106**, by recording a plurality of pieces of data together so that as many of the pieces of data as possible are recorded in the same ECC block (recording timing) by using a cache, it is

5   possible to effectively utilize an unrecorded area in the spare area. Particularly, a Metadata Bitmap or a Logical Volume Integrity Descriptor may not be updated every time a file is recorded. By recording the Metadata Bitmap or the Logical Volume Integrity Descriptor after a plurality of files have been recorded, it is possible to effectively use an unrecorded

10   area of the spare area.

Figure **5** shows an optical disc information recording/reproduction system **500** according to the present invention. The information recording/reproduction system **500** includes a system controller **510**, a drive

15   apparatus **520** for reading and writing information from and onto an optical disc, and an input/output bus **530**.

Between the system controller **510** and the drive apparatus **520**, instructions and responses using a command set and transfer of the

20   read/write data are performed through the input/output bus **530**.

The system controller **510** includes a controller **511** and a memory **512**. The system controller **510** may be a personal computer. The controller **511** may be, for example, a semiconductor integrated circuit

25   such as a CPU (Central Processing Unit) and performs the method described in the embodiments of the present inventions.

Further, a program for causing the controller **511** to perform the method described in the embodiments is stored in the memory **512**. In the

30   controller **511**, a file system, a utility program, or a device driver may be performed.

The drive apparatus **520** includes a system LSI **521**, a

memory 522 and a drive mechanism 523.   A program for causing the system LSI 521 to perform the method described in the embodiments of the present inventions may be stored in the memory 522.   The system LSI 521 may be formed on a semiconductor chip and may include a micro processor.

5

The drive mechanism 523 includes a mechanism for loading an optical disc, a pickup 524 for writing/reading the data from/onto a disc, a traverse mechanism for moving the pickup 524.   The drive mechanism 523 is controlled by the system LSI 521.

10

As explained above, the data of a file can be recorded without being overwritten, some of metadata can be recorded using pseudo-overwrite. Typically, the size of metadata needed to update a file is smaller than the data size of the file, and then the size of data to be overwritten can be reduced.   The size of a file entry is 2048 bytes, and the size of a directory depends on the number of files and the length of the file name. As an example, if each file name is 12 characters and 39 files are recorded in the directory, the directory information can be recorded within a sector of 2048 bytes.   Therefore, the read/write operation can be realized basically for a write-once disc by combining the new recording method and the drive apparatus with the overwritten function.

(Embodiment 2)

25

This embodiment describes a further recording method to write a file onto a write-once disc on which the state is explained in embodiment 1 as Figure 2.

In Figure 2, the unrecorded area is only 2 sectors of SA #m+5 and #m+6, therefore, if the Metadata Bitmap and the root directory are written in the spare area, a file entry of the root directory, and a file entry of the file can not be written any more.   Thus, an additional file can not be recorded, even if the Metadata Bitmap indicates available areas in the

Metadata file, because the unrecorded area in the spare area is used up.

So, hereinafter a recording method to record the file with checking the size of unrecorded areas in the spare area is described.

Figure 3 is a diagram illustrating a configuration of areas on the disc on which the same data as shown in Figure 2 is written.   In Figure 3, as a spare area with the larger size has been provided by a logical formatting operation, the spare area has a larger unrecorded area than that of Figure 2.

Figure 6 is a diagram illustrating data transfer with commands between a drive apparatus and a system controller.   Specific commands can be applied to standards defined by ANSI (American National Standards Institute) or Multi-Media Command Set Standards defined by the INCITS (Inter National Committee for Information Technology Standards) T10.

Steps **S601**, **S603**, **S605**, and **S607** indicate procedures performed by the system controller.   Steps **S602**, **S604**, **S606**, and **S608** indicate procedures of the drive apparatus.

In step **S601**, the type of a medium loaded into the drive apparatus is requested from the system controller and the system controller recognizes the media is the write-once pseudo-overwritable disc and recognizes that the drive apparatus supports the pseudo-overwrite function.

In step **S602**, the drive apparatus reads out information of the type of the loaded disc.   The drive apparatus also determines whether or not the pseudo-overwrite function is supported with respect to the disc. The drive apparatus informs the system controller of these pieces of information.

In step **S603**, by requesting the track information of a write-once

disc, the system controller obtains the information from the drive apparatus. Specifically, the size of an unrecorded area in Track #2, and the next writable address in the track or the last recorded address in the track are requested.   In order to write the data of the file, it is necessary to get the

5   above-described information in advance, and it is checked whether or not the unrecorded area has a prescribed size or more.   Since in Track #2 additional Metadata file may be allocated, for example, the prescribed size may be 128MB for the disc of a whole capacity with 23GB.   If the size of the unrecorded area is less than the prescribed size, the disc is used as a

10   read-only disc.   If the size is equal to or greater than a prescribed size, the procedure goes to the next step.   The size of a file entry is 2 KB.   When only file entries are recorded in an available area of 128MB, file entries corresponding to 65,536 files at most can be recorded.

15   In step **S604**, the drive apparatus reads the information related with the number of tracks, the positional information and the open/close state of each track, or the last recorded address information, from the lead-in area, DMA, or TDMA of the loaded disc.   The system controller is informed of these pieces of information.

20

                In step **S605**, the drive apparatus is instructed to read the Metadata Bitmap area.   As a result, the system controller obtains the Metadata Bitmap, and it is checked whether or not there are available sectors.   When there are available sectors, the procedure goes to the next

25   step.   When there are no available sectors, an additional area for a Metadata file is assigned in the unrecorded area of Track #2 to reserve the available sectors.   In this checking, the system controller may determine whether the additional area for a Metadata file is reserved or not, by using the prescribed size for the available sectors, for example, the prescribed

30   size may be 128KB.

                In step **S606**, the drive apparatus reads data from the specified area, and transfers the data to the system controller.

In step **S607**, by requesting spare area information to the drive
apparatus, the system controller obtains the information and checks
whether or not there is an unrecorded area in the spare area having a
prescribed size or more.

For example, when the size of the unrecorded area is equal to or
greater than 8 MB, the disc is usable as a recordable disc.   When the size
is less than 8 MB, the disc is used as a read-only disc.   The spare area is
used not only for pseudo-overwrite, but also for defect management.
Therefore, an additional unrecorded area is needed to recover defective
sectors on the disc.

In step **S608**, the drive apparatus reads the number of spare
areas, the size of spare areas, and the size of an unrecorded area in each
spare area, from the lead-in area, DMA, or TDMA of the loaded disc.   The
system controller is informed of these pieces of information.

As described above, the data is recorded in the unrecorded area
in the spare area by pseudo-overwrite or defect management.   The write-
once disc drive apparatus of the present invention has a function to send
the free area information as well as the type information of a medium to a
system controller, because the drive apparatus stores the data at some
location which may be different from the location the file system expects.
By requesting this free area information whenever a file is recorded, the
system controller decides whether the file can be recorded or not, as a
result, the system controller can record a file correctly with the related data
in the Metadata file.

(Embodiment 3)

In the previous embodiment 2, the spare area with sufficient size
has to be assigned at the time of formatting on a write-once disc.   However,

a user cannot know how many files and the size of the files that will be recorded on the disc, therefore, it is difficult to decide the appropriate size of the spare area at the time of formatting.   If all of the spare area is used, no file can be recorded on the disc, even if the unrecorded area remains in

5   the user data area.   On the other hand, if a larger spare area is assigned, after all of the user data area is used, unrecorded area may remain in the spare area.

Further, a file system driver has to check the size of the

10   unrecorded area in the spare area each time when a file is recorded, hence the space management becomes difficult to implement.   This is not suitable for the implementation of file system driver of computer systems.

So, hereinafter a recording method in which the direction to

15   replace the data not only within the spare area but also within the user data area is explained.

At first, an idea of the present invention is described:

20   Devices handle the overwriting of existing data by writing the new data to the next writable block and creating an entry in a remapping table stored by the drive apparatus.   The file system continues to use the same logical block number, and the drive apparatus remaps the request to the new location based on the entry in the table.   In order to reduce the size of

25   this table, the file system does not reuse blocks after they are freed.   That is, the file system has to be aware that it is using write-once media, and adjust its behavior accordingly.

The device uses the normal volume space to store the remapped

30   data.   That is, it is writing to the next writable location within the same track that the original block exists.   The file system queries the device for the next writable block whenever it needs to allocate new space.   So both the file system and the device are sharing the same space for writes.

Secondly, the effectiveness of the above idea is described using
Figures **7A-7D, 8A-8B** and **9**:

5   Figures **7A-7D** and **8A-8B** show the blocks in a user data area.
Herein, the block is used rather than the sector to explain the idea generally.
The user data area is recognized as a volume space by the file system.
For each block, a Physical Block Address (hereafter described as PBA) and
a Logical Block Address (hereafter described as LBA) are assigned so that
10   the correspondence between PBA and LBA are decided in advance,
wherein as an example, PBA is assigned from the number 100 and LBA is
assigned from the number 0.

Figure **9** shows the data structure of the remapping table stored
15   by the drive apparatus.   The table has entries, each of which specify the
original address and remapping address.   This data structure may be the
common data structure with the defect list which is used for defect
management for rewritable discs.

20   Typically, it had seemed that the above idea was not effective,
because there are contradictions when applied for rewritable discs.

As shown in Figure **7A**, a rewritable disc has a spare area, in
which a PBA is assigned form the number 200, for example.   In a usual
25   case for a rewritable disc, if the data are written to the blocks of LBA 2 and
5 and these blocks are defective, these data are stored into the blocks PBA
200 and 201 in the spare area using a linear replacement algorithm.   This
means LBA 2 and 5 is re-assigned to PBA 200 and 201.   Thus, when
some block in the volume space becomes an unusable block due to defect,
30   it is compensated with a good block in the spare area.

If the above idea would be applied to a rewritable disc, as shown
in Figure **7B**, the data to write LBA 2 and 5 are stored into the blocks in a

user data area, for example, PBA 103 and 107.   However, these remapped
blocks of PBA 103 and 107 cannot be used to store the data requested to
LBA 3 and 7, as these blocks are replaced as LBA 2 and 5.   This situation
breaks the assumption to provide a defect free logical space, in which the

5   assumption that the data capacity on the rewritable disc shall not be
reduced, when any data is recorded by the file system.   Further, in this
situation, the drive apparatus could not decide the location to remap the
data, because the file system may write the data randomly and only the file
system handles the space bitmap which specifies the available area for

10   recording.

       As explained embodiment 1, the linear replacement algorithm can
also be applied to write-once discs.   As shown in Figure 7C, the spare
area is assigned for the write-once disc in advance.   If the block is

15   overwritten or can not be written due to a defect, the block is compensated
with the block in the spare area.   As examples, when the data D1 is written
to LBA 2, if the block is defective, the block is compensated with the block
of PBA 200.   When LBA 5 is written by the data D2, even if the PBA 105 is
already recorded, the data is stored into PBA 201.

20

       It is supposed that a block to be overwritten and a defective block
should be compensated with another block which belongs to outside of the
volume space.   This idea seems to be a contradiction not only for
rewritable discs, but also for write-once discs.

25

       However, according to the present invention, write-once discs
may not guarantee to provide defect free logical space like rewritable discs,
because if a block on a write-once disc is written once, it is not possible to
change the data in the block.   And the new file system for write-once discs

30   would write the data using sequential recording.   On this point, this idea is
effective for a write-once disc as shown in Figure 7D.   The data D1, D2
and D3 are written to LBA 0, 1 and 2, sequentially, at this time if the block
PBA 102 is defective, then the data may be written to the next block PBA

103.   Further, the data D4, D5 and D6 are written to LBA 4, 5 and 6, sequentially, and then the updated data D5' may be overwritten to LBA 5. On this overwrite, the data is stored into the block PBA 107 which is the next writable location.   Thus, this idea does not require changing the assignment of a logical block number, because whenever the overwritten is needed, the data can be written to next writable location until the user data area is used up.   In case of a rewritable disc, the data can not be recorded into the block used for remapping, but there is no problem for the present invention of a write-once disc.   For example, the data may be recorded further into the block of LBA 7.   In this case, the data is stored in PBA 108 and the entry to specify this remapping from the original address of PBA 107 to the remapping address of PBA 108 is added in the remapping table.

As explained above, according to the present invention, even if the Logical Block Address is double booking, the drive apparatus can record the data by assigning a new Physical Block Address to NWA.

Another important point to be practical to the above idea is to save the entries stored in the remapping table by querying the next writable address from the file system to the drive apparatus.   Figures 8A and 8B show the mechanism to reduce the size of the table.   In this figure, blocks PBA 100 to 105 are recorded in advance.   As shown in Figure 8A, when the data D1 is written to LBA 1, the data is stored in next writable location PBA 106 and one entry is added to specify PBA 101 is remapped to PBA 106.   At this moment, the file system does not know where the data is remapped.   If file system instructs to write the new data D2 to the previous next writable location LBA 6, the data is stored to the next block PBA 107 and one entry is added.   In the present invention, the file system checks the updated next writable address before any data is recorded barring overwrite, and instruct to write the data at the updated next writable address, of which is LBA 7 in Figure 8B.   Thus, an additional entry is not needed.   The file system will not also reallocate the data to the deleted file area, barring the requirement to overwrite for the same reason.

Furthermore, the space management by the file system can be simplified, because the file system uses only NWA in each track to allocate the new area for recording.   This means the file system may not be

5   checked in the spare area, and may not record the space bitmap, especially the Metadata Bitmap.

The unit to remap the area may be an ECC block which consists of plural physical sectors.   When a physical sector is remapped, all of the

10   physical sectors in the ECC block in which the physical sector belongs to are remapped.   In this case, the original address and remapping address in the entry of the remapping table are specified by the physical address of the start sector of each ECC block.   As an addressable unit, the physical block and the logical block may be the physical sector and the logical sector.

15   Even if one sector is newly written, one ECC block including the sector is written, then the NWA is moved to the start sector of the next ECC block. Therefore, when several data are written, these data are allocated so that these data are written with the same ECC block.

20   By the present invention, the merits to use the Metadata Partition can be provided for the use on the write-once disc.   At first, the data to be overwritten is remapped within the track for metadata writing, then the access to retrieve the metadata can be localized and the performance is improved.   As usual, the overwritten data is stored in the same track as

25   long as the track has unrecorded sectors.   When the track is used up by data recording, the overwritten data may be stored into the other track, as it is given the priority to write the data to the other unrecorded physical block in the same track.   Secondly, a Metadata Mirror File can be recorded to improve the robustness, when an additional track for a Metadata Mirror File

30   is assigned.

Herein, the potential for overwriting is estimated for the example case that the capacity of the disc is 23GB (= 23 x 1024^3 bytes), the ECC

block consists of 32 sectors, and the sector size is 2KB (=2 × 1024 bytes). If the average size of the files recorded on the disc is 128KB and 10 files are stored in the directory on average, about 188,000 files and 18,800 directories can be recorded on the disc. When the file system clusters the

5   file entries to be updated into an ECC block, about 6,400 entries are needed in the remapping table. The size of the table becomes about 50KB when the size of the entry is 8 bytes. When the drive apparatus can handle 256KB of the remapping table as the maximum size, 188,000 files and 18,800 directories would be written randomly. Thus, this invention is

10   also practical for the next generation write-once optical discs using blue laser technology.

An example applying the above idea to a new file system based on UDF and the next generation write-once disc is described.

15

Figure **10** is a diagram illustrating a configuration of areas to explain the above mentioned remapping. The area layout explained in the embodiment 1 is also used in this embodiment.

20   Defect Management Area (DMA) is an area in which the defect list is recorded and Temporary DMA (TDMA) is an area, in which a temporary defect list is recorded. The remapping table may be recorded in DMA and TDMA with the defect list and the temporary defect list. Herein, the defect list and the temporary defect list may be used to specify both the

25   replacement information by defect management and the remapping information using the same data structure. It will help to simplify the implementation of the drive apparatus, because the interpretation of the entry is common for defect management and for pseudo-overwrite. When the entry specifies the remapping information, the entry in the table

30   indicates the correspondence between the address of a block to be remapped and the address of a remapped block.

Spare areas are assigned out of a volume space, and the

addresses in the spare area are indicated by SA's (Spare area Address).

The volume space comprises three tracks, in each of which the data is recorded sequentially. The start address of the unrecorded area in the track is managed as Next Writable Address (NWA). The status of a track is described by new terms: "Used" and "Reserved" in order to indicate the data in these tracks can be overwritten and the data may be remapped within the reserved track. The used track means that all sectors in a track have been used for data recording. The reserved track means that there is a sector(s), which has not been recorded. In other words, the data can be incrementally written into the reserved track. Since the volume structure has been previously recorded, Tracks #1 is a used track. Track #2 is a reserved track assigned for metadata recording. Track #3 is a reserved track assigned for user data recording.

An exemplary procedure for updating Data-A file and recording Data-B file onto the write-once optical disc is described.

At first, the Metadata file FE is read to obtain the area allocated for the Metadata file, wherein MA (Metadata file Address) indicates a relative address within a Metadata file.

When the file system updates the Data-A file, the file entry is read in the memory and the file entry is updated in order to register the information to specify the location where the updated data will be written and the related information (e.g., size and updated time, etc) in the memory. The file system instructs the drive apparatus to overwrite the data of Data-A file logically. Then the drive apparatus stores the data (Data-A') physically to the NWA, as the corresponding physical sector is already recorded and the drive apparatus adds the entry to the defect list. If the file system instructs to write the data to NWA after querying the drive apparatus, it is not required to add the entry in the defect list. In this invention, this manner to save the entry is recommended, because the location of the data

- 24 -

can be registered in the file entry.   However, when a part of the large file
has to be updated, the data to be updated may be overwritten instead of
writing the whole data of the file.   After the data is written on the disc, the
file system instructs to overwrite the file entry (Data-A' FE) logically in order
5   to specify the location of the written data and the updated time.   Then the
drive apparatus writes the data physically into the sector shown at MA #k+1
and stores the entry into the defect list.

When the file system newly records the Data-B file under the root
10   directory, the directory is read into the memory and the directory is updated
to add the new file (Data-B file) to this directory in the memory.
Beforehand the file entry of the Data-B file is created in the memory, the
data of Data-B file is written at the NWA in Track #3, and then the file entry
(Data-B FE) and the root directory on the memory are written from the NWA
15   shown as MA #k+2 in Track #2.   To specify the new location of the root
directory, the updated file entry for the directory is instructed to overwrite
and the data is written at the NWA shown as MA #k+4.   Thus, only the file
entry of the directory among metadata is overwritten to save the entry, and
the other metadata (the file entry of the file and the directory) and the data
20   of the file are written without overwriting.

To indicate the integrity of the file structure, the updated Logical
Volume Integrity Descriptor is instructed to overwrite, and the data is written
at the sector SA #m in the spare area.
25

In the above pseudo-overwrite operation, the data to be
overwritten may be stored within the spare area or the NWA in the reserved
track in response to an instruction to write the data into an already recorded
area.   The destination to store the data may be decided by the drive
30   apparatus.   Similarly, the data may be replaced within the spare area or
the NWA in the track by defect management, when the data can not
recorded on the sector due to defect.

In case of rewritable discs, the file system reuses the available area. However, in the present invention the available area is not reused when the available area is used once. For example, in Figure **10**, although the logical sector at MA #i+1 in which the root directory was written became the available sector, the file system will not allocate any data at MA #i+1. If the data were written again into MA #i+1, the data would be remapped. In the case of the data being written on an ECC block basis, one ECC block is written, even if only one sector of the unrecorded ECC block is instructed to be written and the invalid data is recorded in the other sectors of the ECC block. The file system also will not reuse the area where the invalid data is recorded, for the same reason.

Hereinafter a write procedure of the present invention is described. This write procedure is performed by the optical disc information recording/reproduction system described in Figure **5**.

Figures **15A-15D** each comprise a diagram illustrating the blocks in a user data area, when the data is written using the pseudo-overwrite method. The user data area, PBA and LBA are the same definition as described in Figure **7**.

The procedure to update the file is described in Figures **15A** and **15B**. As shown in Figure **15A**, before the file is updated, the blocks PBA 100, 101, and 102 are recorded. The file entry (D3) of the file to be updated is stored in the block LBA 2. As shown in Figure **15B**, the file system;

     1) instructs to read the file entry from LBA 2 in advance,

     2) queries the NWA to the drive apparatus,

     3) instructs to write the updated data (D4) of the file into the NWA (LBA 3), so that the data is written without overwriting, and creates the file entry (D3') so that the positional information in the read file entry is changed to specify the area where the updated data is written, and then

     4) instructs to write the updated file entry (D3') into LBA2 so that

the file entry is overwritten.

The procedure to record the file under the directory is described
in Figure **15C** and **15D**. As shown in Figure **15C**, the blocks PBA 100, 101,
5   102, 108 and 109 are recorded, and two tracks are reserved. The file
entry (D1) of the directory and the directory (D2) are stored in blocks   LBA
0 and 1. As shown in Figure **15D**, the file system;
1) instructs to read the file entry of the directory and the directory
from LBA 0,
10   2) instructs to read the directory from LBA 1,
3) queries the NWA of each track to the drive apparatus in order
to decide the location to write the data without overwriting,
3-1) creates the file's file entry (D6) which has the positional
information of the area allocated for the data of the file,
15   3-2) creates the directory (D7) so that the new file is
registered in the read directory (D2) and
3-3) creates the file entry (D1') of the directory so that the
positional information in the read file entry (D1) is changed to specify
the area where the updated directory is allocated,
20   4) instructs to write the data (D5) of the file into NWA in Track #2
(LBA 10),
5) instructs to write the file entry (D6) of the file from the NWA in
Track #1 (LBA 3)
6) instructs to write the directory (D7) to the next address so that
25   the data is written continuously, and then
7) instructs to write the updated file entry (D1') into LBA0 so that
the file entry is overwritten.

Thus, before any data is written, the file system queries the NWA.
30   This is because the NWA may be changed, as the drive apparatus would
write the data with remapping on its own. Next, the file system provides an
instruction to write the data from NWA so that the data is written without
being overwritten. Then, the file system instructs that the updated data be

written with overwriting.   This order to write the data is important, to save the entry in the remapping table.

In the case of Figure 15B, if the write procedure was not in this order, namely the data D3' is overwritten before the data D4 is written, the data D3' would be written into PBA 103 and the NWA is changed.   This means the data D4 would be written with overwriting into LBA 3, but the data would be remapped into PBA 104.   In the case of Figure 15D, if the write procedure were not in this order, an additional entry would be added in the remapping table, as well.

In the case of rewritable discs, there are no such requirements. Typically, a different order is used for the rewritable disc cases to improve the reliability, considering the recovery when a write procedure may terminate accidentally.   On the other hand, in the case of write-once disc such recovery is not important; rather, it is important to save the entry in the remapping table, because the previous state remains by the write-once feature.

Figures 11A and 11B each comprise a flowchart illustrating a procedure to write the data on a write-once optical disc.   This write procedure is performed by the optical disc information recording/reproduction system described in Figure 5.

When the file is updated, the file system will instruct to write the data of the file so that the data is written without overwriting, and the file system will create the file entry by updating the written file entry and will instruct to write the file entry so that the file entry is overwritten.

When a new file is recorded under the directory, the file system creates the file entry of the file and creates the directory and the file entry of the directory by updating the written directory and it's file entry in the memory and instructs to write the data of the file, the file entry and the

directory so that these data are written without overwriting.   Then the file system will instruct to write the file entry of the directory so that the file entry is overwritten.

5   Figure **11A** shows the steps involved in the procedure to write the data on a write-once optical disc.   A program is used to implement the procedure and is executed by controller **511** contained in the system controller **510**.   The system controller **510** instructs the drive apparatus **520** to write the data on a write-once optical disc.   The controller **511** may

10   include a semiconductor integrated circuit.   The program is provided in various manners.   For example the program may be provided in a form of a computer readable medium having the program recorded thereon. Alternatively, the program may be provided by downloading the program from a server via the internet.   Once the program is installed into a

15   computer, the computer functions as the system controller **510**.

           Figure **11B** shows the procedure performed by the drive apparatus **520**.

20   In step **S1101**, the system controller **510** receives a write request which specifies at least data for a file to be written from the user.   For example, the request is to replace Data-A file as new data or to copy Data-B file from the other media to under the root directory on this disc.   The data to be written is transferred to the memory **512** in the system controller

25   **510** and the destination to record the data is indicated by the pathname in the directory tree structure.

           In step **S1102**, the controller **511** instructs the drive apparatus **520** to read the metadata for managing the directory or file which is requested

30   by user.   The metadata such as file entry and directory are read from the Metadata File.

           In step **S1103**, the controller **511** queries a next writable address

NWA, which indicates a location at which data is to be written next, to the drive apparatus **520** before it instructs the drive apparatus **520** to write the data, because there is a possibility that the drive apparatus **520** will move the NWA. The drive apparatus **520** sends the queried NWA to the system

5   controller **510** and the system controller **510** sends the NWA   to the controller **511**.

In step **S1104**, in response to the user write request, the controller **511** creates or updates at least a part of the metadata so that the

10   data amount to be overwritten is minimized by distinguishing the type of the data to be written.   As described in the explanation for Figure **10**, in the case of a file is updated, the file system creates the file entry to specify the location of the data to be written.   This file entry is the data to be overwritten.   In the case a new file is recorded under the directory, the file

15   system creates the file entry of the new file and updates the directory to register the new file and the file entry of the directory.   The file entry of the directory is the data to be overwritten and the other metadata is the data which can be written without being overwritten. As examples of the updated metadata, the updated metadata may include a file entry of a directory

20   under which the file is recorded, or a file entry of the file

In step **S1105**, the controller **511** instructs the drive apparatus **520** to write the data, which is not needed to be overwritten, at a location indicated by the NWA.

25

In step **S1106**, the controller **511** instructs the drive apparatus **520** to write at least a part of the updated metadata at the location from which the metadata was read so that the data is overwritten on the logical sector. In step **S1105** and **S1106**, the same write command can be used to instruct

30   to write the data, because the drive apparatus **520** can determine whether the data should be written with overwrite or without by checking the status of the logical sector in step **S1111**. In steps **S1105** and **S1106** data is written to different addresses which are indicated in different manners, for example,

step S1105 writes the data specified by a user write request (step S1101), while set S1106 writes the metadata which has been read (step S1102). To avoid possible errors step S1106 is performed after step S1105.

5   In step S1111, the drive apparatus 520 receives the write command, which specifies at least a logical sector in which the data is to be written, from the controller 511 and determines whether the physical sector to which the logical sector corresponds in advance is recorded or not.   In this check, the drive apparatus 520 judges the state of the physical sector
10   using the logical sector number which is instructed to write by the controller 511.   When the logical sector number is smaller than the NWA, the physical sector is recorded, or else unrecorded.   If the physical sector is unrecorded, go to S1112, else (if recorded) go to S1113.

15   In step S1112, the drive apparatus 520 writes the data to the physical sector that corresponds to the logical sector in advance.

           In step S1113, the drive apparatus 520 writes the data to the other unrecorded physical sector.   When the data is remapped on an ECC
20   block basis, the other unrecorded physical sector is one of the sectors belonging to the ECC block which is the next writable block.   For example, in case the ECC block consists of 32 sectors, the second sector in the ECC block is instructed to be overwritten, the data is physically written into the second sector in the next writable ECC block.   Thus the relative address
25   within the ECC block is kept for the remapped ECC block.

           In step S1114, the drive apparatus 520 creates the entry as the remapping information to specify the original address related with the physical sector corresponding to the logical sector in advance and the
30   remapping address related with the physical sector in which the data is written, and stores the remapping information to the defect list/remapping table.   In case the ECC block consists of 32 sectors, the original address is the start address of the ECC block to which the original physical sector

belongs and the remapped address is the start address of the ECC block to which the remapped physical sector belongs.

When this invention is applied to the recording/reproduction system such as a consumer video recorder or a consumer video player, the file system and the drive apparatus may be controlled by the common micro processor as shown in Figure **12**.   In this case, the file system may not query the NWA to the drive apparatus, because the recording/reproduction system knows the NWA.   At first, the NWA in each track is checked when the disc is loaded in the drive unit, then this system can manage the NWA after some data is written.

Figure **12** shows an optical disc information recording/reproduction system **1200** which is a part of a consumer video recorder or a consumer video player.   The information recording/reproduction system **1200** includes a controller **1211**, a memory **1212** and a drive mechanism **523** for reading and writing information from and onto an optical disc.   The controller **1211** may be, for example, a semiconductor integrated circuit such as a CPU (Central Processing Unit) and performs the method described in the embodiments of the present inventions.   Further, a program for causing the controller **1211** to perform the method described in the embodiments is stored in the memory **1212**.   In the controller **1211**, a file system, a utility program, or a device driver may be performed. The drive mechanism **523** may be controlled by the controller **1211**.

Figure **13** is a flowchart illustrating a procedure to write the data on a write-once optical disc by the recording/reproduction system explained in Figure **12**.

In step **S1301**, the controller **1211** receives the request from the user.   For example, the request is to replace Data-A file as new data or to copy Data-B file form the other media to the root directory on this disc.

The data to be written is transferred to the memory **1212** and the destination to record the data is indicated by the pathname in the directory tree structure.

5   In step **S1302**, the file system read the data to retrieve the directory or file which is requested by user.   The metadata such as file entry and directory are read from the Metadata File.

           In step **S1303**, the file system gets the NWA before writing the
10   data.

           In step **S1304**, in response to the user requests, the file system creates some of the metadata so that the data amount to be overwritten is minimized by distinguishing the type of the data to be written.   As
15   described in the explanation for Figure **10**, in the case of a file being updated, the file system creates the file entry to specify the location of the data to be written.   This file entry is the data to be overwritten.   In the case of a new file being recorded under the directory, the file system creates the file entry of the new file and updates the directory to register the
20   new file and the file entry of the directory.   The file entry of the directory is the data to be overwritten and the other data is the data which can be written without being overwritten.   The data type is distinguished as type I: no need to be overwritten and type II: needs to be overwritten.

25   In step **S1305**, the file system instructs to write the type I data into the logical sector from the NWA. Then, go to step **S1311** which is the operation to write the data of type I.

           In step **S1306**, the file system instructs to write the type II data
30   into the logical sector.   Then go to step **S1321** to perform the operation to overwrite the data in the logical sector.   The write command instructed to write the data may be different for type I and II.

In step **S1311**, the data is written to the physical sector which corresponds to the logical sector in advance.

In step **S1321**, the data is logically overwritten in the logical sector and physically written into the other unrecorded physical sector, especially to the NWA.

In step **S1322**, the entry as the remapping information is created to specify the original address of the physical sector corresponding to the logical sector in advance and the remapping address of the physical sector in which the data is written, and is stored to the defect list.

The reproduction procedure from the write-once disc on which disc the data is written using the above explained methods Figures **11A**, **11B** and **13** is explained hereinafter. Figure **14** is a flowchart illustrating a procedure to read the data from a write-once optical disc by the recording/reproduction system explained in Figures **5** and **12**. Wherein, the recording /reproduction system receives a read instruction which specifies at least a logical sector from which data is to be read.

In step **S1401**, the original addresses of all entries stored in the remapping table are searched. As the logical sector number to be read is instructed, the physical sector number which corresponds to the logical sector in advance is used to search the entry which indicates the physical sector is remapped. When the remapping is performed on an ECC block basis, the start address of the ECC block is registered in the entry. In this case, it is checked whether the physical sector number belongs to the remapped ECC block or not. This operation may be done by a drive apparatus in case of an Information recording/reproduction system as described in Figure **5**.

In step **S1402**, if the entry is found, go to step **S1403**, else step **S1404**.

In step **S1403**, the original address of the physical sector is replaced with the remapping address of the physical sector, which is indicated in the found entry.   If the entry is not found in step **S1402**, then the address to be read is determined as the address of the physical sector corresponding to the logical sector specified by the read instruction.   If the entry is found in **S1402**, then the address to be read is determined as the remapping address corresponding to the original address found in the remapping table.

In step **S1404**, the data is read at the determined address.

As explained above, when the address is not found, the data is read from the physical sector corresponding to the logical sector in advance, because the remapping table shows the data stored in the logical sector is not remapped to the other physical sector.

When the address is found, the data is read from the physical sector specified by the remapping address in the found remapping information, because the entry found in the remapping table shows the data stored in the logical sector is remapped.

If the data is remapped within the volume space, especially to the NWA in the track rather than into the spare area, the data can be read out more quickly.

As described in the above embodiments, this invention can be applied to the drive apparatus on the assumption that the file system minimizes the data amount to be overwritten and the file system instructs to write the data to unrecorded sector by querying the drive apparatus, when the data is written without being overwritten.

(Embodiment 4)

In this embodiment, as an example, guidelines and requirements are described for the UDF file system implementation.

(Benefits to Using A Pseudo-Overwrite Method)

In order to reduce complexity due to physical characteristics, new sequential recording media with overwritable features are introduced as pseudo-overwritable media.   The pseudo-overwrite method is applied to this pseudo-overwritable media.   The following sets forth some of the benefits of introducing the new sequential recording media:

1) An overwritable volume space and a defect free space are provided, similar to a rewritable media.   In other words, compatibility due to write-once media is ensured by a drive unit which supports an overwritable mechanism and defect management.

2) Session close and Border close are not necessary, as a read-only drive unit supporting Pseudo-overwrite media has access to an unrecorded area.

3) Metadata Partition and its mirror can be used.

(Characteristics of Pseudo-Overwritable Media)

The physical characteristics of pseudo-overwritable media are described from the viewpoint of what is required by the file system driver.

Pseudo-overwritable media supports multi-track recording and an overwritable function for the logical sectors in a volume space.   More than one track can be used to record.   A new track can be assigned as a reserved track.   A sequential recording mode is to be used within a track, to simplify the space management.   The pointer to the recordable area is Next Writable Address (NWA), which is obtained through an inquiry of a drive apparatus.

When the data is intended to be recorded on the recorded logical sector, the data is recorded either within the Spare Area by a linear replacement algorithm, or to NWA within the volume space.   As a benefit of remapping the data to NWA within the volume space, all available media capacity can be used, even if all of the Spare Area is recorded.   A read modify write operation is also supported, therefore, each logical sector can be overwritten separately.

The address information where the data is replaced or remapped from the original address is stored as a defect list entry of volume space and managed by a drive apparatus.

(Write Strategy by the file system)

In general, a track can be assigned by considering the data type to be recorded; for example, metadata or specific data such as audio/visual data, still picture data, music data and so forth.   On multi-track recording, when a reserved track is used up, a new track is assigned, adaptable to the amount of recorded data.   The track for metadata recording is explicitly indicated as an extent of a Metadata File or a Metadata Mirror File.

If pseudo-overwrite media has the restriction that a new track can not be assigned after a track is assigned at the end of the volume space, multi-track recording is used in an intermediate state in which only one AVDP is recorded at LSN 256.   According to ECMA 167 requirements, the AVDP shall be recorded in at least two of three locations (i.e. LSN 256, LSN – 256 and Last LSN).   In this example, the track is assigned to record AVDP at Last LSN - 256 or Last LSN.

A Metadata Mirror File can be also used.   It is recommended to create a Metadata Mirror File when the disc is stored for archiving.   In case of online usage, the implementation is to record such that the contents recorded in a Metadata File and a Metadata Mirror File have the same

offset in each file, although the offset of NWA in each track sometimes may not be the same due to the recording condition of each track.

5   Figure **16A** shows an example of a track layout after a logical format.   A first volume structure including AVDP at LSN 256 and the related file structure is recorded.   Then, a track is assigned for metadata recording, and the metadata is recorded in the track.   The remaining area in the volume space is used for file data recording.

10   Figure **16B** shows an example of a track layout after some files are recorded.   After the first track is used up, a new track is assigned. Thus, additional tracks may be assigned one after another.

      Figure **16C** shows another example of a track layout after some
15   files are recorded.   When the track for metadata recording is used up, an additional area for Metadata File may be allocated in the data track as an extent, not a track.

      (Requirements for File System)
20

      Requirements for pseudo-overwrite method are listed as follows:
      1) Implementations are to recognize Pseudo-overwritable media by inquiring to the drive unit.
      2) An Unallocated Space Bitmap and an Unallocated Space Table
25   shall not be recorded.
      3) A Metadata Bitmap File shall not be recorded.
      4) Implementation should query for NWA in each track prior to writing additional data.   If a write command is issued to an already-recorded area, the defect list entry is used.   Therefore, this requirement is important to
30   reduce an unnecessary defect list entry, as the size of defect list is limited.
      5) Deleted blocks should not be reused to query NWA for the same reason.
      6) The metadata to be overwritten should be minimized. It is

recommended that only the directory File Entry should be overwritten.

INDUSTRIAL APPLICABILITY

5   The present invention is useful to provide a recording method for a write-once disc using a logical overwritable mechanism, and a semiconductor integrated circuit for use in the recording apparatus or the reproduction apparatus.

10